

UN ALGORITMO SAEM PARA EL PROBLEMA DE COMPLETACIÓN DE MATRICES

*Anaís Frangeline Acuña Sosa ** Jhonny Otilio Escalona Pérez

Recibido: 23/03/2015 Aprobado: 04/07/2015

Resumen

En este trabajo estudiamos el problema de completación de matrices. Se presenta en diversas áreas como la teoría de sistemas y control, procesamiento de imágenes y filtrado colaborativo. Considerando un modelo de factorización probabilística de matrices, establecemos una propuesta basada en estadística Bayesiana y un algoritmo Maximización Espectativa (EM) estocástico para recubrir una matriz de datos a partir de una muestra de sus entradas. El método propuesto no requiere de parámetros de regularización y da un estimado del rango de la matriz, en contraste con el método de Factorización Probabilística Bayesiana de Matrices (BPMF). Los resultados muestran que el algoritmo propuesto da mejores estimados del rango de la matriz en comparación con un algoritmo basado en lagrangeanos aumentados y es más eficiente que el método BPMF.

Palabras clave: Completación de matrices, algoritmo EM, algoritmo SAEM, filtrado colaborativo, análisis de componentes principales.

* *Departamento de Investigación de Operaciones y Estadística, Decanato de Ciencias y Tecnología, Universidad Centroccidental Lisandro Alvarado, Barquisimeto, Venezuela, MSc, anaís.frangeline@gmail.com*

** *Departamento Investigación de Operaciones y Estadísticas, Decanato de Ciencias y Tecnología, Universidad Centroccidental Lisandro Alvarado, Barquisimeto, Venezuela, MSc, jhonnyescalona@ucla.edu.ve*

A SAEM ALGORITHM FOR MATRIX COMPLETION PROBLEMS

Abstract

In this work we deal with matrix completion problem. This problem arise in different fields, for example, systems and control theory, image processing and collaborative filtering. Given a probabilistic matrix factorization model, we present an approach based on Bayesian statistics and a stochastic expectation maximization algorithm to retrieve an array of data from a sample of its inputs. The proposed method does not requires regularization parameter and estimates the rank of the matrix, in contrast to the BPMF method. Our results show that the proposed method outperforms to an augmented lagrangian algorithm and the BPMF method in its ability to find the rank of the matrix and in efficiency respectively.

Keywords: Matrix completion, EM algorithm, SAEM algorithm, collaborative filtering, principal components analysis.

Introducción

En el problema de completación de matrices tenemos una matriz de datos $F \in \mathbb{R}^{N \times M}$, la cual queremos conocer con la mayor precisión posible, pero solo conocemos un subconjunto de sus entradas. Un modelo de factorización probabilística de matrices, en el cual utilizan un algoritmo basado en métodos Monte Carlo con cadenas de Markov (MCMC) para aproximar la distribución predictiva que permite completar la matriz F es propuesto por Salakhutdinov y Mnih, y aunque este método mostró tener buenos resultados para el problema de Netflix, es costoso computacionalmente (Salakhutdinov y Mnih, 2008b).

Considerando el modelo de factorización probabilística, en el trabajo antes citado, se propuso resolver problemas de completación de matrices, encontrando dos de ellas $U \in \mathbb{R}^{D \times N}$ y $V \in \mathbb{R}^{D \times M}$ tales que: $F = U'V$. Se considera que las filas de la matriz F están formadas por muestras de un vector aleatorio proveniente de una distribución normal. Esto permite a través de la implementación de un algoritmo EM para poblaciones normales y considerando un análisis de componentes principales obtener la matriz $V \in \mathbb{R}^{D \times M}$ deseada. Para la estimación de la matriz $U \in \mathbb{R}^{D \times N}$ se diseña un algoritmo SAEM

acoplado con un algoritmo MCMC; dado que en la práctica el algoritmo SAEM a mostrado ser menos costoso computacionalmente, comparado con algoritmos similares.

El algoritmo EM, propuesto por Dempster, Laird, y Rubin (1977) es una herramienta muy utilizada en estadística para encontrar el estimador de máxima verosimilitud (o máximo a posteriori) de parámetros en modelos probabilísticos, es de especial utilidad cuando se tienen observaciones incompletas. El algoritmo alterna entre un paso de esperanza (paso E), donde se calcula el valor esperado de la función de verosimilitud mediante la inclusión de las variables no observadas como si fueran observables, y un paso de maximización (paso M) donde se calculan los estimadores de máxima verosimilitud de los parámetros del modelo, mediante la maximización del valor esperado de la verosimilitud del paso E. Los parámetros que se encuentran en el paso M se usan para comenzar el paso E siguiente, y así el proceso se repite, hasta alcanzar la convergencia.

En los casos donde el paso E del algoritmo EM no se puede o es difícil de calcular, algunos autores proponen usar el algoritmo SAEM (Delyon, Lavielle, y Moulines, 1999), el cual reemplaza el usual paso E del algoritmo EM por un procedimiento de aproximaciones estocásticas, se demuestra que bajo supuestos muy generales el algoritmo SAEM converge al máximo (local o global) de la función de verosimilitud de las observaciones.

El algoritmo SAEM comprende tres pasos, un paso de simulación, donde se generan muestras de la distribución a posteriori de los datos no observados, dado los datos observados; un paso de estimación a través aproximaciones estocásticas y un paso de maximización. En muchas situaciones prácticas no es posible generar directamente las muestras de la distribución a posteriori de los datos no observados, por lo que el paso de simulación puede realizarse a través de un algoritmo de método Monte Carlo con cadenas de Markov (MCMC). Con esta modificación en el paso de simulación, se demuestra que bajo ciertas condiciones el algoritmo SAEM acoplado con un algoritmo MCMC converge al estimador de máxima verosimilitud (Kuhn y Lavielle, 2004). Los algoritmos MCMC permiten obtener pseudo-muestras provenientes de una distribución de probabilidad, constru-

yendo una cadena de Markov en el espacio de parámetros.

En la próxima sección detallamos el problema de completación de matrices. En la tercera sección, presentamos el modelo de factorización probabilística de matrices. En la cuarta sección, exponemos los detalles del método propuesto. Posteriormente, en la quinta sección mostramos resultados de la implementación del método, exhibiendo comparaciones con uno o dos métodos diferentes para dos tipos de problemas. En la sexta sección, concluimos con algunos comentarios finales.

Problema de completación de matrices

Las matrices que tratamos en este problema suelen recibir el nombre de matrices parciales. Dada una matriz $A \in \mathbb{R}^{n \times m}$, se dice que A es una matriz parcial si parte de sus elementos son conocidos y el resto están por especificar (Hassan, 2009).

Cuando damos valores a los elementos no especificados de una matriz parcial A , obtenemos una matriz convencional, que recibe el nombre de completación de A . Así, el problema de completación de matrices consiste en encontrar una completación de una matriz parcial dada; pero de tal forma que ésta verifique ciertas propiedades o condiciones deseadas.

Este problema ha sido estudiado desde diversos puntos de vista, por ejemplo, a través de optimización convexa; demostrándose que bajo algunas condiciones puede ser resuelto (Candès y Recht, 2009), considerando el problema de minimización del rango y encontrando una solución minimizando la norma nuclear; dado que una hipótesis muy común es la de suponer que la matriz posee rango bajo o aproximadamente bajo. Algunos algoritmos basados en esta estrategia, requieren en cada iteración calcular la descomposición de valores singulares de la matriz, trayendo consigo un alto costo computacional.

Se ha propuesto un método alternante para encontrar la matriz de rango bajo, evitando la descomposición de valores singulares, emplean un modelo de factorización y construyen un algoritmo no lineal de sobre-relajación sucesiva (Wen, Yin, y Zhang, 2012). Este algoritmo se modificar agregando información de dualidad en cada iteración

a través del uso de un método de Lagrangeano aumentado (Lara, Oviedo, y Yuan, 2014), obteniendo un nuevo algoritmo alternante de tipo Lagrangeano. Estos métodos de Lagrangeano aumentado son una herramienta empleada en algunos problemas de optimización, existe otro método de Lagrangeano aumentado para la recuperación exacta de matrices de rango bajo (Chen, Lin, y Ma, 2010) .

Desde la perspectiva Bayesiana han surgido también diversas propuestas, una de estas es basada en un modelo Gaussiano y un algoritmo MAP-EM (Léger, Yu, y Sapiro, 2010). Por otro lado, existe un modelo de factorización probabilística de matrices no lineal con procesos Gaussianos (Lawrence y Urtasun, 2009). Ambas propuestas tratan en específico el problema particular de completación de matrices para filtrado colaborativo.

Modelo de factorización probabilística de matrices

Supongamos que tenemos una matriz parcial, $F \in \mathbb{R}^{N \times M}$ con entradas conocidas F_{ij} con $(i, j) \in \Omega$, donde $\Omega \subset \{(i, j) | 1 \leq i \leq N, 1 \leq j \leq M\}$.

Salakhutdinov y Mnih (2008b) proponen un método de factorización de rango bajo, como ha sido propuesto a través de optimización convexa, pero desde un enfoque probabilístico. Esta factorización probabilística de matrices, es un modelo probabilístico lineal con ruido Gaussiano. La idea detrás de esta propuesta consiste en encontrar dos matrices $U \in \mathbb{R}^{D \times N}$ y $V \in \mathbb{R}^{D \times M}$, tales que $F = U'V$.

Sean U_i, V_j con $i \in \{1, 2, \dots, N\}$ e $j \in \{1, 2, \dots, M\}$, vectores columna correspondientes a la i -ésima columna de U y j -ésima columna de V , respectivamente. En este modelo se define la distribución condicional sobre F_{ij} con $(i, j) \in \Omega$, y las distribuciones de los prior sobre $U \in \mathbb{R}^{D \times N}$ y $V \in \mathbb{R}^{D \times M}$ como:

$$p(F|U, V, \alpha) = \prod_{i=1}^N \prod_{j=1}^M [N(F_{ij}|U_i'V_j, \alpha^{-1})]^{I_{ij}}, \quad (1)$$

$$p(U|\mu_U, \Sigma_U) = \prod_{i=1}^N N(U_i|\mu_U, \Sigma_U), \quad (2)$$

$$p(V|\mu_V, \Sigma_V) = \prod_{j=1}^M N(V_j|\mu_V, \Sigma_V), \quad (3)$$

donde $N(x|\mu, \Sigma)$ denota la distribución normal con media μ y matriz de covarianza Σ , I_D denota la matriz identidad de dimensión D e I_{ij} es una función indicadora la cual es igual a 1, si la entrada F_{ij} es conocida e igual a 0, en otro caso.

En la exposición de Salakhutdinov y Mnih (2008a) se considera $\mu_U = 0$, $\mu_V = 0$ y $\Sigma_U^{-1} = \alpha_U^{-1} I_D$, $\Sigma_V^{-1} = \alpha_V^{-1} I_D$, matrices diagonales. La estimación en este modelo se realiza mediante la maximización del log-posterior sobre U y V con hiperparámetros fijos:

$$\ln p(U, V|F, \alpha, \alpha_V, \alpha_U) = \ln p(F|U, V, \alpha) + \ln p(U|\alpha_U) + \ln p(V|\alpha_V) + C, \quad (4)$$

donde C es una constante que no depende de los parámetros.

La maximización de (4) es equivalente a minimizar la función de la suma de los errores al cuadrado con términos de regularización cuadráticos:

$$E = \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^M I_{ij} (F_{ij} - U_i' V_j)^2 + \frac{\lambda_U}{2} \sum_{i=1}^N \|U_i\|_{Fro}^2 + \frac{\lambda_V}{2} \sum_{j=1}^M \|V_j\|_{Fro}^2, \quad (5)$$

donde $\lambda_U = \alpha_U/\alpha$, $\lambda_V = \alpha_V/\alpha$ y $\|\cdot\|_{Fro}^2$ denota la norma Frobenius. Un mínimo local de la función definida en (5) se puede encontrar usando algún método de optimización basado en gradientes sobre U y V .

En este modelo es necesario controlar la complejidad manualmente, buscando valores apropiados de los parámetros de regularización λ_U y λ_V . Basados en este inconveniente, posteriormente (Salakhutdinov y Mnih, 2008a) proponen un tratamiento totalmente Bayesiano del modelo de factorización probabilístico de matrices (BPMF). En este caso se consideran priors Gaussianos-Wishart para

los hiperparámetros, tomando $\Theta_U = (\mu_U, \Sigma_U)$ y $\Theta_V = (\mu_V, \Sigma_V)$:

$$\begin{aligned} p(\Theta_U | \Theta_0) &= p(\mu_U | \Sigma_U) p(\Sigma_U) \\ &= N(\mu_U | \mu_0, (\beta_0 \Sigma_U)^{-1}) W(\Sigma_U | W_0, \nu_0), \end{aligned} \quad (6)$$

$$\begin{aligned} p(\Theta_V | \Theta_0) &= p(\mu_V | \Sigma_V) p(\Sigma_V) \\ &= N(\mu_V | \mu_0, (\beta_0 \Sigma_V)^{-1}) W(\Sigma_V | W_0, \nu_0), \end{aligned} \quad (7)$$

donde $\Theta_0 = \{\mu_0, \beta_0, \nu_0, W_0\}$ y W es la distribución de Wishart con ν_0 grados de libertad y matriz de escala W_0 de dimensión $D \times D$.

La distribución predictiva de F_{ij}^* , es obtenida por la marginalización sobre los parámetros e hiperparámetros del modelo:

$$p(F_{ij}^* | F, \Theta_0) = \int \int p(F_{ij}^* | U_i, V_j) p(U, V | F, \Theta_U, \Theta_V) p(\Theta_U, \Theta_V | \Theta_0) d\{U, V\} d\{\Theta_U, \Theta_V\} \quad (8)$$

Como la evaluación exacta de la distribución predictiva es analíticamente intratable debido a la complejidad del posterior, debemos recurrir a inferencia aproximada. Los algoritmos MCMC pueden ser usados para aproximar la distribución predictiva de la ecuación (8) a través de:

$$p(F_{ij}^* | F, \Theta_0) \cong \frac{1}{T} \sum_{t=1}^T p(F_{ij}^* | U_i^{(t)}, V_j^{(t)}), \quad (9)$$

donde las muestras $\{U_i^{(t)}, V_j^{(t)}\}$ son generadas mediante la ejecución de una cadena de Markov cuya distribución estacionaria es la distribución posterior a través de los parámetros e hiperparámetros del modelo $\{U, V, \Theta_U, \Theta_V\}$.

En Salakhutdinov y Mnih (2008a) se propone utilizar un muestreador de Gibbs para encontrar las muestras deseadas y aproximar la distribución predictiva (8), mediante (9).

Método propuesto

Dada una matriz parcial $F \in \mathbb{R}^{N \times M}$, con entradas conocidas F_{ij} con $(i, j) \in \Omega$. Con el método propuesto deseamos encontrar dos matrices $U \in \mathbb{R}^{D \times N}$ y $V \in \mathbb{R}^{D \times M}$ tales que: $F = U'V$.

Consideremos que las filas de la matriz F están formadas por N muestras de un M -vector aleatorio, proveniente de una distribución normal con media μ y matriz de covarianza Σ ; dado que F es una matriz parcial tenemos un problema de datos incompletos. La implementación del algoritmo EM para poblaciones normales nos permite calcular los estimadores de máxima verosimilitud $\hat{\mu}$, $\hat{\Sigma}$ de μ y Σ , respectivamente.

Con la estimación $\hat{\Sigma}$ de Σ es posible obtener la matriz $V \in \mathbb{R}^{D \times M}$ deseada. Tomando V' como una matriz cuyas columnas corresponden a los D vectores propios asociados a los D mayores autovalores $\lambda_1, \dots, \lambda_D$ de $\hat{\Sigma}$, por análisis de componentes principales tenemos la siguiente relación:

$$U' = F V', \quad (10)$$

donde $VV' \approx I_D$ y el valor D es escogido de tal forma que $\lambda_1 + \dots + \lambda_D$ represente el porcentaje de varianza que se desea capturar.

De este modo, solo resta estimar la matriz $U \in \mathbb{R}^{D \times N}$. Considerando el modelo PMF dado por las ecuaciones (1), (2) y (3), donde U_1, \dots, U_N se considera una muestra vectorial, suponemos que U_1, \dots, U_N constituyen un conjunto de vectores aleatorios estadísticamente independientes con $U_i \sim N(\mu_{U_i}, \Sigma_{U_i})$, $\forall i \in \{1, \dots, N\}$. Esto permite implementar de forma sencilla un algoritmo SAEM para cada U_i y estimar $\hat{\mu}_{U_i}$, $\hat{\Sigma}_{U_i}$, los estimadores de máxima verosimilitud de μ_{U_i} y Σ_{U_i} , respectivamente.

En este caso, la distribución condicional sobre F_{ij} con $(i, j) \in \Omega$ se puede escribir como:

$$\mathcal{L}_{U_i}(F) = \prod_{j=1}^M [N(F_{ij} | U_i' V_j, \alpha^{-1})]^{I_{ij}}, \quad (11)$$

donde $N(x | \mu, \sigma^2)$ denota la distribución normal con media μ y varianza σ^2 , I_D denota la matriz identidad de dimensión D e I_{ij} es una función indicadora la cual es igual a 1, si la entrada F_{ij} es conocida e igual a 0, en otro caso.

Como la distribución de U_i con $i \in \{1, \dots, N\}$, pertenece a la familia exponencial, el paso de aproximación del algoritmo SAEM para estimar los parámetros de máxima verosimilitud de μ_{U_i} y Σ_{U_i}

de cada U_i , se reduce a actualizar sus estadísticos suficientes. Por otra parte, el paso de maximización consiste en actualizar los estimadores de máxima verosimilitud, en función de las actualizaciones de los estadísticos suficientes.

El algoritmo SAEM propuesto se presenta en el Cuadro 1.

Puede ser necesario considerar una regularización de las matrices $\hat{\Sigma}_{U_i}^{(k)}$, para $k \geq 1$; haciendo:

$$\hat{\Sigma}_{U_i}^{(k)} = \hat{\Sigma}_{U_i}^{(k)} + \hat{\epsilon} I_D, \quad (12)$$

donde $\hat{\epsilon}$ es una constante pequeña. Esto garantiza que cada matriz $\hat{\Sigma}_{U_i}^{(k)}$ sea de rango completo.

Debemos notar que este algoritmo será implementado para cada U_i con $i \in \{1, \dots, N\}$, es decir, es utilizado para estimar una a una las columnas de la matriz U . Utilizamos para su inicialización la estimación de los parámetros obtenidos de la implementación del algoritmo EM para poblaciones normales y la matriz $V \in \mathbb{R}^{D \times M}$ ya obtenida. El algoritmo se detendrá cuando se haya encontrado el estimador de máxima verosimilitud de las observaciones, esta convergencia suele ser rápidamente alcanzada. El método propuesto posee la ventaja de no requerir escoger parámetros de regularización como en el método de factorización probabilística de matrices, el cual consiste en encontrar un mínimo local de la función definida en (5), algo que lo hace muchas veces difícil de implementar, y con el cual se obtiene los parámetros iniciales del modelo para implementar el método BPMF.

Experimentos numéricos

En esta sección presentamos algunos resultados de la implementación del método propuesto, para esto se desarrollaron códigos en MATLAB y se realizaron dos tipos de experimentos con matrices simuladas. Comparamos el rendimiento de nuestro método con el método de lagrangeano aumentado inexacto (inexact ALM) (Chen et al., 2010), además del método de factorización probabilística Bayesiana de matrices (BPMF) (Salakhutdinov y Mnih, 2008a).

En la implementación del algoritmo EM para poblaciones normales, se utilizó una tolerancia de $1e-4$ y 500 como número máximo de

Cuadro 1: Algoritmo SAEM

Inicialización:

- Proporcionar una tolerancia $\epsilon > 0$.
- Calcular: $\hat{\mu}_{U_i}^{(0)} = V \hat{\mu}$, $\hat{\Sigma}_{U_i}^{(0)} = V \hat{\Sigma} V'$.
- Hacer: $U_i^{(0)} = \hat{\mu}_{U_i}^{(0)}$.

Para $k \geq 1$, haga mientras $|\mathcal{L}_{\hat{U}_i^{(k-1)}}(F) - \mathcal{L}_{\hat{U}_i^{(k)}}(F)| > \epsilon$:

1. **Simulación:** Realizar $m(k)$ iteraciones de un algoritmo Metropolis-Hastings, para obtener $U_{i1}^{(k)}, \dots, U_{im(k)}^{(k)}$.
2. **Aproximación estocástica:** Actualizar $S(\mu_{U_i})^{(k-1)}$ y $S(\Sigma_{U_i})^{(k-1)}$, con:

$$S(\mu_{U_i})^{(k)} = S(\mu_{U_i})^{(k-1)} + \gamma_k \left(\sum_{j=1}^{m(k)} U_{ij}^{(k)} - S(\mu_{U_i})^{(k-1)} \right),$$

$$S(\Sigma_{U_i})^{(k)} = S(\Sigma_{U_i})^{(k-1)} + \gamma_k \left(\sum_{j=1}^{m(k)} U_{ij}^{(k)} U_{ij}^{(k)'} - S(\Sigma_{U_i})^{(k-1)} \right),$$

donde $\{\gamma_k\}_{k \geq 1}$ es una sucesión decreciente de tamaños de pasos positivos, la cual converge a 0 y satisface que $\sum_k \gamma_k = \infty$.

3. **Maximización:** Actualizar $\hat{\mu}_{U_i}^{(k-1)}$ y $\hat{\Sigma}_{U_i}^{(k-1)}$, con:

$$\hat{\mu}_{U_i}^{(k)} = \frac{1}{m(k)} S(\mu_{U_i})^{(k)},$$

$$\hat{\Sigma}_{U_i}^{(k)} = \frac{1}{m(k)} S(\Sigma_{U_i})^{(k)} - \hat{\mu}_{U_i}^{(k)} \hat{\mu}_{U_i}^{(k)'}$$

4. Hacer: $U_i^{(k)} = \hat{\mu}_{U_i}^{(k)}$.

iteraciones. El rango de la descomposición es seleccionado considerando el porcentaje de varianza que se desea capturar. Luego de obtener la matriz V deseada procedemos a estimar la matriz U , donde una iteración del algoritmo SAEM utilizado para estimar una columna de esta matriz, realiza 100 iteraciones del algoritmo Metropolis-Hastings. Consideramos una tolerancia de $1e - 4$ y un número máximo de iteraciones de 25, las cuales no suelen ser alcanzadas dada la rápida convergencia. Tanto en el método propuesto como en BPMF, la varianza del ruido es fijado en $\alpha^{-1} = \frac{1}{2}$.

Experimentos numéricos de tipo 1

Para realizar estos experimentos creamos matrices parciales de forma aleatoria por el siguiente procedimiento:

1. Generamos dos matrices aleatorias $U \in \mathbb{R}^{D \times N}$ y $V \in \mathbb{R}^{D \times M}$ con entradas independientes e idénticamente distribuidas provenientes de una distribución normal estándar.
2. Hacemos $F = U'V$.
3. Borramos aleatoriamente alrededor del 50 % de las entradas de la matriz F , para obtener una matriz parcial \tilde{F} .

Luego de la estimación calculamos el error relativo como:

$$\frac{\|F - \hat{F}\|_{\mathcal{F}}}{\|F\|_{\mathcal{F}}}, \quad (13)$$

donde \hat{F} es una completación para la matriz parcial \tilde{F} y $\|\cdot\|_{\mathcal{F}}$ denota la norma de Frobenius.

En el Cuadro 2, mostramos los resultados del cálculo del error relativo (Error Rel.), el rango seleccionado de la descomposición (Rg. est.) y el tiempo de ejecución en segundos (T. Ej. (seg)) para el método propuesto, BPMF e inexact ALM.

En los resultados de BPMF en la columna correspondiente a Rg. est., hemos empleado el símbolo “-” para indicar que el método no realiza la estimación, este valor debe ser fijado por el experimentador y en sus implementaciones hemos usado el valor conocido de D . Por su

Cuadro 2: Comparación entre el método propuesto, BPMF e inexact ALM en experimentos numéricos de tipo 1.

	N	M	D	Error Rel.	Rg. est.	T. Ej. (seg)
M. propuesto	50	30	5	0,2652	5	41,8364
BPMF	50	30	5	1,1494	-	12,1142
inexact ALM	50	30	5	0,4804	29	0,3378
M. propuesto	100	80	30	0,6739	21	220,8104
BPMF	100	80	30	1,2111	-	87,8891
inexact ALM	100	80	30	0,5482	61	1,5338
M. propuesto	180	120	60	0,8779	36	281,0380
BPMF	180	120	60	1,2461	-	489,6364
inexact ALM	180	120	60	0,6031	120	2,6289

parte, el método propuesto realiza el cálculo del valor D , considerando que el porcentaje de varianza que se desea capturar es de 95 %.

Podemos observar en el Cuadro 2 que el error con el método BPMF es siempre mayor, mostrando al método no eficaz con esta clase de matrices parciales (las cuales poseen entradas no enteras y algunas negativas). En relación al método propuesto e inexact ALM, sus errores relativos pueden ser considerados cercanos, pero es importante destacar que los valores de Rg. est., obtenidos con el método propuesto son más cercanos al verdadero valor D .

Experimentos numéricos de tipo 2

En estos experimentos crearemos de forma aleatoria matrices parciales, simulando problemas de filtrado colaborativo. Este problema comprende datos acerca de las calificaciones dadas a ciertos artículos, pero ya que los usuarios solo califican un subgrupo de estos, generalmente muy pocos, la matriz de preferencias posee pocas entradas, así el problema radica en completar esta matriz para que el vendedor pueda hacer recomendaciones a sus usuarios. Por lo general, cada fila de la matriz de preferencias representa un usuario y cada columna un artículo.

Para crear las matrices parciales de estos experimentos, seguiremos el siguiente procedimiento:

1. Generamos aleatoriamente matrices $\tilde{F} \in \mathbb{R}^{N \times M}$ con entradas enteras de r_{\min} a r_{\max} y solo alrededor del 30 % de sus entradas.

2. Seleccionamos por cada fila una calificación conocida, con estas calificaciones formamos un subconjunto Ω de las entradas conocidas de \tilde{F} . Al conjunto Ω lo llamaremos conjunto de prueba.
3. Borramos las entradas de la matriz \tilde{F} pertenecientes al conjunto Ω , para obtener una nueva matriz parcial F .

Luego de realizar la estimación, se evalúa el rendimiento del método en el conjunto de prueba, a través del cálculo del error absoluto medio normalizado (NMAE), definido como:

$$NMAE = \frac{1}{(r_{\max} - r_{\min}) |\Omega|} \sum_{(i,j) \in \Omega} |F_{ij} - \hat{F}_{ij}|, \quad (14)$$

donde \hat{F} es una completación para la matriz parcial F .

En los experimentos hemos considerado $r_{\min} = 1$ y $r_{\max} = 5$. En el Cuadro 3, presentamos los resultados de la implementación del método propuesto y de BPFM, para matrices parciales $F \in \mathbb{R}^{N \times M}$ generadas según el último procedimiento descrito. En la tercera columna se encuentran los errores absoluto medio normalizado (NMAE) para cada caso, y en la última columna los tiempos de ejecución correspondientes, en segundos (T. Ej. (seg)).

Cuadro 3: Comparación entre el método propuesto y BPFM en experimentos numéricos de tipo 2.

	N	M	NMAE	T. Ej. (seg)
M. propuesto	50	30	0,2950	14,5384
BPFM	50	30	0,3900	73,1588
M. propuesto	120	100	0,2729	40,2140
BPFM	120	100	0,3000	326,4987
M. propuesto	250	180	0,2800	516,2469
BPFM	250	180	0,2570	1,2523e + 003
M. propuesto	500	350	0,2820	3,8892e + 003
BPFM	500	350	0,2565	6,2269e + 003

En estos experimentos, el rango de la descomposición para el método BPFM es fijado en 30 para las matrices parciales de orden 50×30 , para las matrices de orden 120×100 y 250×180 en 60, y para

las de orden 500×350 es fijado en 120. Con el método propuesto el rango de la descomposición es seleccionado considerando que se desea capturar el 99 % de la varianza.

Con los resultados del cálculo de **NMAE**, mostrados en el cuadro 3, se puede notar que el método propuesto presenta mejores resultados en dos de los casos, y aunque por una diferencia pequeña produzca un mayor **NMAE** en los últimos de éstos, debemos considerar que en todos ellos los tiempos de ejecución son considerablemente menores.

Conclusiones

Los resultados del método propuesto se han comparado con dos métodos, mostrándolo competitivo con respecto al método BPMF propuesto por (Salakhutdinov y Mnih, 2008a). Debemos destacar que sus tiempos de ejecución son considerablemente menores, además no requiere escoger parámetros de regularización, algo necesario para implementar el método BPMF. En relación al método inexact ALM (Chen et al., 2010), con el cual realizamos comparaciones en los experimentos numéricos de tipo 1, podemos mencionar que si bien sus tiempos de ejecución son menores y en algunos casos se obtienen mejores errores relativos, con el método propuesto hemos obtenido mejores aproximaciones del rango de la descomposición.

El método propuesto presenta el mayor gasto computacional en la implementación del algoritmo EM para poblaciones normales, algoritmo que nos permite estimar los estimadores de máxima verosimilitud de la media y matriz de covarianza de la distribución normal de la que suponemos provienen las muestras que representan las filas de la matriz parcial F , y luego nos permite obtener la matriz V deseada. Como estudios futuros relacionados a este trabajo, se puede considerar el estudio de métodos o técnicas que mejoren este proceso de estimación.

Referencias

Candès, E., y Recht, B. (2009). Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*,

- 9(6), 717-772.
- Chen, M., Lin, Z., y Ma, Y. (2010). The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. *arXiv preprint arXiv:1009.5055*.
- Delyon, B., Lavielle, M., y Moulines, E. (1999). Convergence of a stochastic approximation version of the em algorithm. *The Annals of Statistics*, 27(1), 94-128.
- Dempster, A., Laird, N., y Rubin, D. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1), 1-38.
- Hassan, R. (2009). *Problemas de completación de matrices parciales* (Tesis Doctoral no publicada). Universidad Politécnica de Valencia.
- Kuhn, E., y Lavielle, M. (2004). Coupling a stochastic approximation version of em with an mcmc procedure. *ESAIM: Probability and Statistics*, 9, 115-131.
- Lara, H., Oviedo, H., y Yuan, J. (2014). Matrix completion via a low rank factorization model and an augmented lagrangean succesive overrelaxation algorithm. *CompAMa*, 2(2), 21-46.
- Lawrence, N., y Urtasun, R. (2009). Non-linear matrix factorization with gaussian processes. En *Proceedings of the 26th annual international conference on machine learning* (pp. 601-608).
- Léger, F., Yu, G., y Sapiro, G. (2010). Efficient matrix completion with gaussian models. *arXiv preprint arXiv:1010.4050*.
- Salakhutdinov, R., y Mnih, A. (2008a). Bayesian probabilistic matrix factorization using markov chain monte carlo. *Proceedings of the International Conference in Machine Learning. ACM*, 880-887.
- Salakhutdinov, R., y Mnih, A. (2008b). Probabilistic matrix factorization. *Advances in neural information processing systems*, 1257-1264.
- Wen, Z., Yin, W., y Zhang, Y. (2012). Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm. *Mathematical Programming Computation*, 4(4), 333-361.